



Five Technology Pitfalls for Nonprofits: Finding Cost-Effective Solutions

Are you
making any of
these expensive
mistakes?



By David Gleason

Finding technology to help meet your goals can be challenging on a limited budget. But take heart! It is possible to have reliable, high-performing systems without breaking the bank. How? By heeding these five cautions:

1. Don't depend too heavily on used equipment

Many nonprofits are saddled with unreliable technology — albatross donations that seemed like a good idea at the time but turned out to be costly to maintain. How many of us have waited for an older computer to reboot over and over? A clear policy on what systems to accept as donations will help keep maintenance costs down and productivity high.

Used equipment can be helpful, but only under certain circumstances. Computers generally go obsolete in three to five years. Machines that have a single, dedicated use may last longer, but after seven years, maintaining any office computer will cost more (in time and expense) than buying a new one. After five years, most machines are useful only for training technicians, simple Web browsing, or student use (note that

since security software slows older systems down, even kids will become impatient). An organization will get at most two years of use out of a three-year-old donation.

It's best to look at the life cycle of a machine and determine the "total cost of ownership" (TCO). This is the sum of costs over the system's usable life: purchase price, software, upgrades, maintenance, user support, and so on.

Computers generally go obsolete in three to five years.

The ratio of the TCO to the anticipated life cycle will tell you the system's annual cost. For example, a new computer with a five-year deployment (fully configured with services and software) might have a TCO of \$5,000 over five years, or \$1,000 per year. A used three-year-old machine might have a TCO of \$3,000 over two years (initial configuration can be expensive, especially if the system must be wiped clean, and you must also pay for software licenses) for an annual cost of \$1,500. That's 50% *more*

than the price of a new computer. If the donation were only one year old, the configuration would be cheaper, so the TCO might be \$2,500 over four years, or \$625 per year. This option would be 38% *less* per year than a new computer.

By contrast, cutting-edge technologies have their own costs. While it's generally good to get new computers when possible, a marginally faster machine can cost far more than a new standard workhorse. And, especially with software, it's usually smart to let products prove themselves in the market before you adopt them.

In addition to direct costs, taking what comes instead of acquiring what's needed can slow operational growth. It's wise to resist the impulse to accept just any donation. As an industry, we're prone to undervaluing time. For example, you might think, "I'll just spend an evening getting it running," only to find one evening turning into five evenings invested in a machine of marginal utility.

Here are ways to bring used IT (information technology) equipment into your nonprofit:

- **Establish** a policy for accepting

donations of used equipment.

- **Steer clear** of systems more than two years old unless you have a specific strategic use for them.
- **Run** a TCO analysis on used versus new options.
- **Buy** standard configurations whenever possible.
- **Pay close attention** to security. Don't let a used computer bring a virus into your office.

2. Don't underbudget for IT infrastructure

Many nonprofits believe that lousy technology is part of the deal, because "money is short and should be entirely devoted to programs." This is a fallacy, the more so since underspending on technology can invite serious security problems.

Wishful thinking never makes systems work well. A nonprofit must design networks and choose software with focused objectives in mind and clear standards for reliability and support.

Select your IT systems through a careful review of functionality. Use professionals for jobs like wiring, server configuration, firewall installation, and software customization. Finally, solicit expert advice *before* you spend money on infrastructure. It's useful to have an advocate who understands both your organizational mission and the technologies under consideration.

"Inexpensive" IT support can lead to inadequate protection from Internet threats, such as spyware, viruses, and other attacks. In severe cases, these omissions can result in lost data and a realization that there are insufficient backups to recover information.

In the interest of saving money, it can be easy to defer policies around security, passwords, and privacy. Don't do it! Recovering from a serious infection on just one computer can take days.

All organizations need dependable support. A quick response to urgent needs saves money in the long run. For example, timing is critically important in outreach. Labels must be printed and ready the day before volunteers come to stuff envelopes.

These ideas can help you budget properly for IT needs:

- **Find** reliable experts to provide planning, deployment, and maintenance support to your organization.
- **Be sure** you have a disaster recovery plan.
- **Identify** staff with specific expertise, and make that expertise explicit.
- **Document** the relationship of systems to organizational objectives.

3. Don't fail to clarify operational needs

It may seem counterintuitive at first, but good technology has more to do with organizational management than with technology itself. Systems codify policies and procedures. Faulty policies become faulty systems. Furthermore, most "IT projects from hell" arise from a lack of clarity about needs.

This can be especially problematic in nonprofits because we're often so focused on the needs of program beneficiaries that we can lose sight of operational needs. Computers "grease the skids," but they also

continued on page 24

Partner with us,
and we will all keep
making **change**.



Earn a good return on your money while revitalizing communities!

When your nonprofit banks at ShoreBank, you receive competitive rates, cost-effective accounts and responsive customer service from professionals who understand the nonprofit community. Meanwhile, you're conserving natural resources, rehabbing affordable housing, and more.

For more on accounts and services available nationwide call **1.800.669.7725** ext 5507 or visit **www.sbk.com**



SHOREBANK

Let's change the world.®



lock you into specific procedures. For instance, a jobs program might have an opportunity to start providing training services, but the system is set up to track job openings and candidates, not course registrations. Adding such functionality must be understood as part of the new offering's cost. The parameters of the program will determine the functional requirements of the system.

Consider these suggestions to clarify needs:

- **Align** your IT strategy with your organization's mission.
- **Before deploying a system, decide** how you'll know when it's complete and working correctly.
- **Ensure** that your Web site promotes your organization's mission.
- **Avoid** embedding problematic assumptions into the system.
- **Pay close attention** to data quality.
- **Articulate** the purposes of the IT system: fundraising, accounting, communications, outreach, networking, publicizing, educating, and so on.

4. Don't ignore the decisions incorporated in your systems

In most technology, functions are developed and tested, and once they're working reliably, they're subsumed, or incorporated, into larger systems. These larger systems, in turn, are subsumed into still larger ones. Once components, subsystems, and systems are operating, the processes become subsumed – invisible and unavailable to the end user. The failure of subsumed objects can be catastrophic, as in the Challenger and Columbia space shuttle disasters, where once the subsumed errors manifested, lives had already been lost. Nonprofits aren't immune to subsumption problems, although the effects usually aren't fatal. For example, a donor system that has subsumed erroneous dates can publicly embarrass an organization.

To the layperson, technology seems as if it should be extremely malleable. For instance, people tend to think making changes to software should be easy because programming is just a set of instructions, and it's not like renovating a building made up of physical materials. However, the principle of subsumption makes it clear that changing base components is like moving building foundations and can require changes to entire systems. The Y2K problem, for example, was a result of subsumed date processing: No one thought the systems would be around long enough to need four-digit dates.

"Subsumption ethics" means managing the process by which decisions become incorporated into technology and then forgotten. Technological systems, by nature, repeat operations. Even if those operations have unethical impacts within or outside an organization, the systems will continue to execute them anyway.

Here are tips for coping with the subsumption problem:

- **Recognize** that IT systems are more grown than built, and that good management matters far more than specific technologies.
- **Avail yourself** of strategic thinking to grow systems that are reliable and well-suited to your organization.
- **Actively manage** political issues such as board involvement and user satisfaction.
- **Establish** policies you can apply as systems grow in order to incorporate good decisions throughout the process.

5. Don't rely on home-grown software

Computer systems codify operations. It's often tempting to build your own system, because you want to keep costs down and be flexible enough to respond to unforeseen situations. Resist this temptation! A system built by a single employee who leaves can be a debilitating legacy. Long-term system support is as important as clear requirements and reliable infrastructure.

Organizations can become sad-

dled with home-grown software that's difficult to manage and impossible to maintain. This burden can be compounded by the desire to continue to focus on mission and not on technical issues. These "legacy systems" can hang around for a long time.

Customizing software often results in poorly structured data with compromised integrity. In general, it's expensive to gather and maintain data. The entire IT industry struggles with quality, but home-grown applications are more susceptible than packaged systems to mediocre data, obsolescence, and inaccessible information.

Many organizations inherit an application which no one knows how to maintain. It's often better to establish a relationship with a software provider who will keep the system up-to-date and be at the other end of the phone when you have an urgent problem.

Use these recommendations to avoid software and system woes:

- **Take** the build/buy decision very seriously.
- **Limit** your system requirements.
- **Collect** as little data as necessary to achieve your purpose.
- **Evaluate** the availability of needed technical expertise, and fill in the gaps.
- **Run** due diligence on all projects, vendors, and staff to ensure that they have a track record of success at these projects. If they don't, bring in outside expertise. ■

Resources

"Donated Computers—Burden Or Blessing?", *Nonprofit World*, Vol. 21, No. 4.

Piersall, Joanne R., "Contingency Planning: Facing Disaster and Surviving," *Nonprofit World*, Vol. 11, No. 3.

These resources are available at www.snpo.org/members.

David Gleason, MA, PMP (dgleason@info-ethics.com, www.info-ethics.com) is managing consultant of Information Ethics, Inc., improving outcomes of computer technologies for all stakeholders.